IB 04/51799

PCT/IB04/51799

Europäisches Patentamt

European Patent Office

Office européen des brevets

# Bescheinigung

# Certificate

# Attestation

Die angehefteten Unterlagen stimmen mit der ursprünglich eingereichten Fassung der auf dem nächsten Blatt bezeichneten europäischen Patentanmeldung überein.

The attached documents are exact copies of the European patent application described on the following page, as originally filed.

Les documents fixés à cette attestation sont conformes à la version initialement déposée de la demande de brevet européen spécifiée à la page suivante.

**Patentanmeldung Nr. Patent application No. Demande de brevet n°**

03103575.1

Der Präsident des Europäischen Patentamts;
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets
p.o.

R C van Dijk

Anmeldung Nr:
Application no.: 03103575.1
Demande no:

Anmeldetag:
Date of filing: 26.09.03
Date de dépôt:

Anmelder/Applicant(s)/Demandeur(s):

Koninklijke Philips Electronics N.V.
Groenewoudseweg 1
5621 BA Eindhoven
PAYS-BAS

Bezeichnung der Erfindung/Title of the invention/Titre de l'invention:
(Falls die Bezeichnung der Erfindung nicht angegeben ist, siehe Beschreibung.
If no title is shown please refer to the description.
Si aucun titre n'est indiqué se referer à la description.)

A method and apparatus for testing an IC

In Anspruch genommene Prioriät(en) / Priority(ies) claimed /Priorité(s) revendiquée(s)
Staat/Tag/Aktenzeichen/State/Date/File no./Pays/Date/Numéro de dépôt:

Internationale Patentklassifikation/International Patent Classification/
Classification internationale des brevets:

G01R31/303

Am Anmeldetag benannte Vertragstaaten/Contracting states designated at date of filing/Etats contractants désignées lors du dépôt:

AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HU IE IT LU MC NL
PT RO SE SI SK TR LI

A method and apparatus for testing an IC

The invention describes a method and apparatus for X-masking logic for testing of digital integrated circuits. It is beneficial to compact test responses since this allows for reducing test data volume and test time. However, the values of some test responses cannot be determined a priori, while other test responses may be tri-stated or unreliable. These unknown, tri-stated, and unreliable test responses, referred to as 'X' test responses, cause that reliable compaction of test responses becomes very difficult or even impossible. The invention proposes circuitry (X-masking logic) that transforms all 'X' test responses into fixed, known responses and hence enables test response compaction. The X-masking logic consists of a linear-feedback shift register (LFSR), a phase shifter (PS) with weighting logic, and control circuitry to selectively enable or disable masking.

Manufacturing tests for digital integrated circuits (ICs) are typically performed using ATE (automated test equipment), also called IC tester. An ATE stores test stimuli and applies these stimuli to the IC. The subsequent responses of the IC are observed by the ATE. The ATE compares these observed responses with the expected responses of a fault-free IC. This allows to determine whether the IC passed or failed the test, and to diagnose faults. The ATE stores the test stimuli and the fault-free test responses in its memory.

Digital ICs usually contain scan chains. This implies that the flip-flops in the IC can be configured into serial shift registers ('scan chains') in test mode. A test is performed by repeatedly (1) shifting a test stimulus into the scan chains of the IC in test mode, (2) operating the IC for a number of clock cycles in its functional application mode, and (3) shifting out the test responses in test mode. When shifting out the test responses, a new test stimulus is usually shifted in.

Continuous improvements in IC process technology allow for an exponential increase of the numbers of transistors on an IC, according to Moore's law. Consequently, the test data volumes are increasing exponentially too. New IC process technologies introduce new defect types, and additional tests are needed to detect these defects, which causes an even larger increase of test data volumes. This sharp increase causes critical problems related to ATE usage and test costs. The test data volumes may exceed the capacity of ATE memories, and test times are increasing.

Solutions are provided by test data compression techniques and built-in self-test (BIST) techniques. These techniques compact the test responses in space and/or time by using special test response compaction circuitry, like a MISR (multiple-input signature register) or SCL (space-compaction logic). This compaction circuitry may reside either on-chip, off-chip (e.g. on the load-board), or in the ATE. The ATE now only has to store the compacted test responses of a fault-free IC.

The actual problem, for which this invention proposes a solution, is that some test responses have to be discarded (or masked) before the test responses can be compacted. This is the case for tri-stated, unknown, and unreliable test responses. We will refer to such test responses as 'X' responses.

- Tri-stated test responses have a high impedance state ('Z'), and may originate for instance from bi-directional pins or from buses.
- Unknown test responses have an unknown value ('U'), and may originate for instance from uninitialized embedded memories, mixed-signal modules, and conflicting or floating buses. Also when shifting in the first test stimulus into an IC, unknown values are shifted out of the IC. An unknown test response implies a test response that is either 'L' (low, i.e. 0), 'H' (high, i.e. 1), or tri-stated ('Z'), but it cannot be determined a priori (i.e. before the test is performed) what the value will be.
- Unreliable test responses have an unreliable result that might be incorrect. This may for instance happen for circuits with multiple clock domains, where unreliable test responses are generated due to clock skew when testing the signal lines that cross clock-domain boundaries.

Compacting test responses that contain 'X' responses will lead to unreliable compacted test responses that cannot be used to determine whether the circuit passed or failed the test, or for diagnosis of faults. Hence, test response compaction cannot be applied for circuits that produce 'X' responses.

Tri-stated, unknown, and unreliable values in test responses should be discarded, or they should be prevented from occurring.

A partial, state-of-the-art solution is to resolve unknown test responses inside the circuit by adding appropriate DfT (design-for-testability) hardware. For instance, test points may be inserted in signal lines that carry unknown values, which forces the signal lines to a known, fixed value. Adding DfT to the circuit for preventing unknown values however requires (often manual) modification of the circuit. This may be impossible (for instance in case of 'hard cores' for which only the circuit layout is available), or undesirable (for instance if the DfT hardware affects timing, such that the circuit cannot run at the required frequency).

A more preferable solution is to mask all tri-stated, unknown, and unreliable test responses at the output of the circuit, before they enter the compaction circuit. This can be achieved with X-masking logic (XML). The XML masks out all 'X' responses, where 'X' indicates either a tri-stated ('Z'), an unknown ('U'), or an unreliable test response. Masking implies that the 'X' response is replaced by a known, fixed value (for instance 'H'). When implementing the XML on-chip, the additional silicon area for the XML should be small. Furthermore, the XML should be flexible, such that it can be accommodated in case different test responses are obtained, for instance due to different test patterns or due to design changes. The operation of the XML may be controlled from the ATE. In that case, the amount of data stored on the ATE for controlling the XML, should be small. Also, the bandwidth requirements for transporting this data from the ATE to the IC should be small. Various XML implementations have been proposed in prior art, but these XML do not completely meet the previous requirements.

This invention proposes a new XML implementation, as shown in Fig. 1. The figure shows a digital CUT (circuit-under-test) with *n* scan chains. The outputs of the scan chains are connected to a compactor for compacting the test responses. The compactor may for instance be implemented by a MISR (multiple-input signature register) or space compaction logic. The XML is placed between the CUT and the compactor. In Fig. 1, the XML and compactor are placed on the IC as an example. The XML and/or the compactor may also be placed off-chip, for instance on the ATE load-board or inside the ATE.

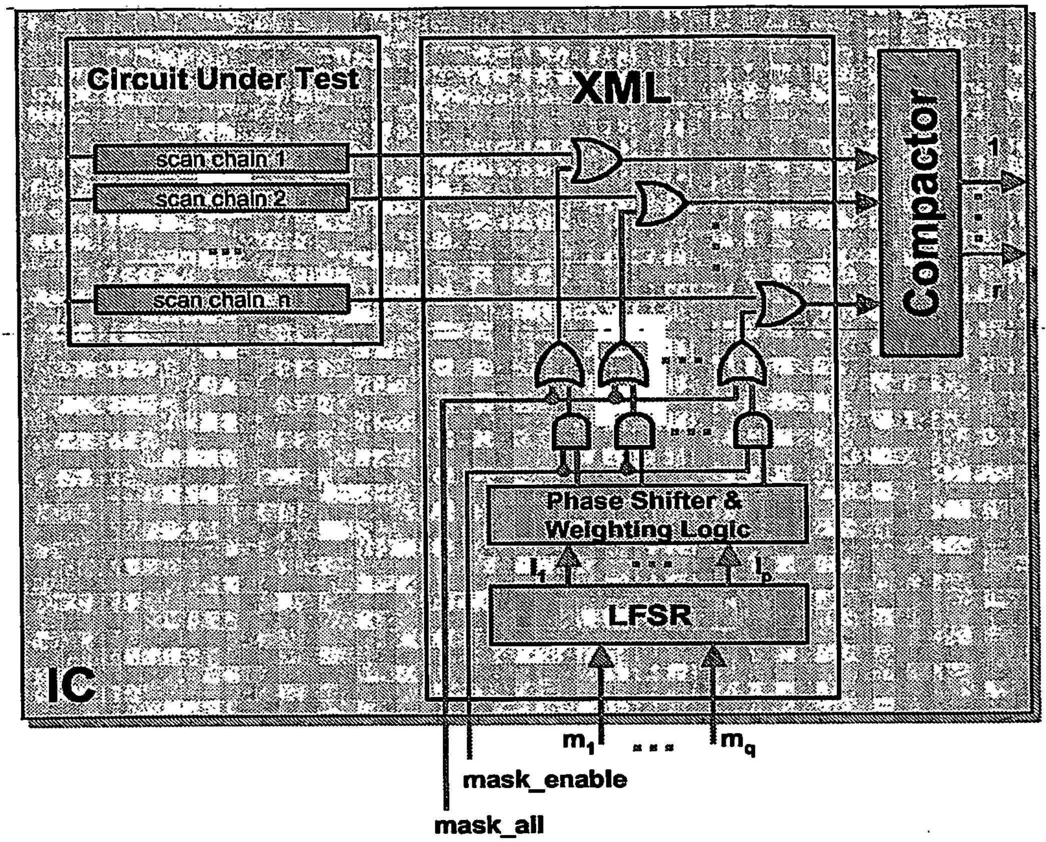

Figure 1: **Circuit with compactor and X-masking logic**

The XML consists of an LFSR (linear-feedback shift register), a phase shifter with weighting logic, and control logic. The XML is controlled from the ATE, using the control signals *mask_all* and *mask_enable*. Furthermore, the ATE updates the LFSR state through the inputs $m_1, \ldots, m_q$.

The proposed XML allows for reducing the amount of masking data that has to be stored on the ATE. In Fig. 1, the ATE has to store $q$+2 bits per clock cycle (for the *mask_all*, *mask_enable*, and $m_1, \ldots, m_q$ signals). These $q$+2 bits are used by the XML to generate an $n$-bit mask, and the XML allows that $q \ll n$. Hence, without XML, the ATE has to store an $n$-bit mask for each clock cycle, and $n$ IC input pins are required to transfer such a mask from the ATE to the IC. When using the XML, the ATE has to store only $q$+2 bits of XML control data for each clock cycle, and only $q$+2 IC input pins are required to transfer this data from the ATE to the IC.

An LFSR is a well-known structure, consisting of a shift register and a feedback network of XOR gates. The LFSR is $p$-bits wide (i.e. its shift register contains $p$ flip-flops). In each clock cycle, the state of the LFSR is (partially) updated using $q$ bits from the ATE, where $q \leq p$, through inputs $m_1, ..., m_q$. This updating can for instance be implemented by adding XOR gates to the shift register, where the input of each XOR gate is connected to $m_l$. The output of the LFSR is used as input to the phase shifter.

A phase shifter is also a well-known structure, consisting of a network of XOR gates. The phase shifter generates $n$ outputs signals from $p$ input signals, where $p \leq n$, and reduces correlation between the output signals. Weighting logic may be added to the phase shifter. This weighting logic allows for biasing the probability that a certain output is 0 or 1. The weighting logic typically consists of some combinational logic, and optionally may be controlled using additional input signals from the ATE.

The $n$ outputs of the phase shifter plus optional weighting logic (PS+WL) are used as inputs to the control circuit. The control circuit in Fig. 1 consists of three stages.

1. The first stage consists of $n$ AND gates, controlled by the *mask_enable* control input. When *mask_enable* = 1, the outputs of the AND gates are equal to the outputs of the PS+WL. When *mask_enable* = 0, the outputs of all AND gates are forced to 0, and hence the $n$-bit mask of the PS+WL is replaced by a mask with all 0's.2. The second stage of the control circuit consists of $n$ OR gates, controlled by the *mask_all* control input. When *mask_all* = 0, the outputs of the OR gates are equal to the outputs of the first stage of the control circuit. When *mask_all* = 1, the outputs of all OR gates are forced to 1, and hence the $n$-bit mask is replaced by a mask with all 1's. The $n$ outputs of the second stage form the actual $n$-bit mask.

3. The third stage of the control circuit consists of $n$ OR gates that are placed at the scan chain outputs of the CUT. The OR gates modify the test responses, as shifted out of the $n$ scan chains, with the $n$-bit mask.

The third stage of the control circuit is always required, since this stage implements the actual masking. It applies the $n$-bit mask to $n$ outputs of the CUT. In Fig. 1, the third stage consists of OR gates. If the output from the second stage to such an OR gate is 1, then the test response from the corresponding scan chain is replaced by a 1, and the test response is said to be masked. If the input is 0, then the test response passes the OR gates, and is not masked. Hence, a value of 1 in a mask indicates that the test response of the corresponding scan chain is masked, and a value of 0 indicates that the test response is not masked. Another embodiment would be that a value of 0 in a mask indicates that the test response of the corresponding scan chain is masked, and 1 that the test response is not masked. In that case, the third stage should contain AND gates instead of OR gates. Hence, it should be clear that the control logic as shown in Fig. 1 is only an example embodiment, and alternative implementations using different types of logic gates are possible as well.

The first stage and/or the second stage of the control logic are optional. The phase shifter and/or the weighting logic are optional too.

The control logic may also be extended. The use of the *mask_all* control signal may be refined. For instance, two such control signals may be used that both allow to mask half of all scan chains. Similarly, the use of the *mask_enable* control signal may be refined. For instance, two such control signals may be used that both enable/disable masking of half of all scan chains.

The operation of the XML is described in more detail in the following example.

**Example**

The simple example in Fig. 2 shows an XML for a CUT with 4 scan chains. The XML includes a 4-bit LFSR, a phase shifter, and a 3-stage control logic.

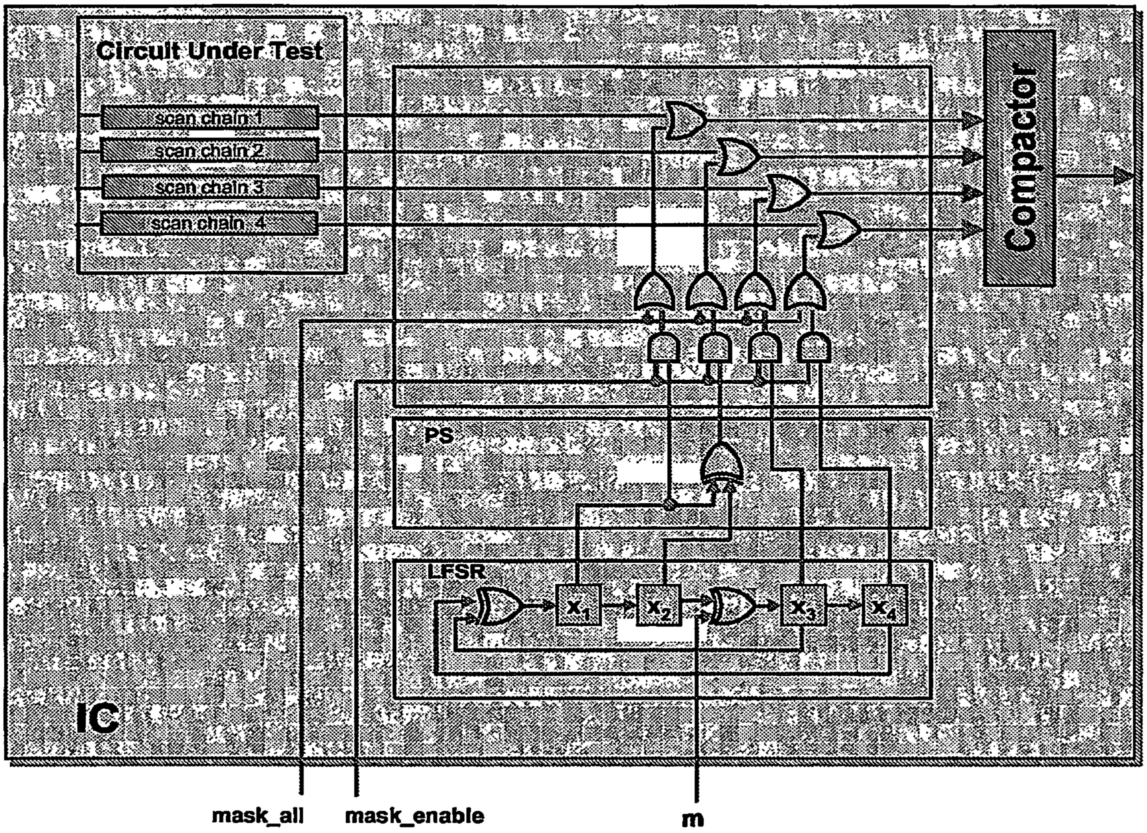

**Figure 2: Example XML for CUT with 4 scan chains**

The symbols $x_1$, $x_2$, $x_3$, and $x_4$ represent the initial value (or 'state') of the LFSR at clock cycle 1. The state of the LFSR in subsequent clock cycles can be expressed in terms of these symbols plus input $m$ from the ATE. These expressions can easily be derived by means of symbolic simulation. The LFSR states in the next 5 clock cycles are as follows, where $LFSR_i$ indicates the LFSR state in clock cycle $i$, and $m_i$ indicates the value of signal $m$ in clock cycle $i$.

$LFSR_1 = (x_1, x_2, x_3, x_4)$

$LFSR_2 = (x_3 \oplus x_4, x_1, x_2 \oplus m_1, x_3)$

$LFSR_3 = (x_2 \oplus x_3 \oplus m_1, x_3 \oplus x_4, x_1 \oplus m_2, x_2 \oplus m_1)$

$LFSR_4 = (x_1 \oplus x_2 \oplus m_1 \oplus m_2, x_2 \oplus x_3 \oplus m_1, x_3 \oplus x_4 \oplus m_3, x_1 \oplus m_2)$

$LFSR_5 = (x_1 \oplus x_3 \oplus x_4 \oplus m_2 \oplus m_3, x_1 \oplus x_2 \oplus m_1 \oplus m_2, x_2 \oplus x_3 \oplus m_1 \oplus m_4, x_3 \oplus x_4 \oplus m_3)$

$LFSR_6 = (x_2 \oplus x_4 \oplus m_1 \oplus m_3 \oplus m_4, x_1 \oplus x_3 \oplus x_4 \oplus m_2 \oplus m_3, x_1 \oplus x_2 \oplus m_1 \oplus m_2 \oplus m_5, x_2 \oplus x_3 \oplus m_1 \oplus m_4)$

In a similar way, the output signals of the phase shifter can be expressed in terms of the symbols $x_1$, $x_2$, $x_3$, $x_4$, and $m_i$.

$PS_1 = (x_1, x_1 \oplus x_2, x_3, x_4)$

$PS_2 = (x_3 \oplus x_4, x_1 \oplus x_3 \oplus x_4, x_2 \oplus m_1, x_3)$

$PS_3 = (x_2 \oplus x_3 \oplus m_1, x_2 \oplus x_4 \oplus m_1, x_1 \oplus m_2, x_2 \oplus m_1)$

$PS_4 = (x_1 \oplus x_2 \oplus m_1 \oplus m_2, x_1 \oplus x_3 \oplus m_2, x_3 \oplus x_4 \oplus m_3, x_1 \oplus m_2)$

$PS_5 = (x_1 \oplus x_3 \oplus x_4 \oplus m_2 \oplus m_3, x_2 \oplus x_3 \oplus x_4 \oplus m_1 \oplus m_3, x_2 \oplus x_3 \oplus m_1 \oplus m_4, x_3 \oplus x_4 \oplus m_3)$

$PS_6 = (x_2 \oplus x_4 \oplus m_1 \oplus m_3 \oplus m_4, x_1 \oplus x_2 \oplus x_3 \oplus m_1 \oplus m_2 \oplus m_4, x_1 \oplus x_2 \oplus m_1 \oplus m_2 \oplus m_5, x_2 \oplus x_3 \oplus m_1 \oplus m_4)$

The outputs of the phase shifter serve as a mask. This mask is updated further in the control logic using the inputs *mask_enable* and *mask_all*, as described in Table 1.

| mask_all | mask_enable | |
|---|---|---|
| 0 | 0 | No scan chain output is masked |
| 0 | 1 | Masking of scan chain outputs is controlled by the LFSR and PS |
| 1 | 0 | All scan chain outputs are masked |
| 1 | 1 | All scan chain outputs are masked |

Table 1: **Control signals**

Assume that the test responses as observed at the outputs of the scan chains in the 6 clock cycles, are as follows:

$R_1 = (\underline{U}, \underline{U}, \underline{U}, \underline{U})$<br>
$R_2 = (L, L, H, L)$<br>
$R_3 = (H, \underline{U}, L, H)$<br>
$R_4 = (\underline{U}, H, \mathbf{H}, L)$<br>
$R_5 = (\mathbf{H}, L, L, \mathbf{H})$<br>
$R_6 = (L, \mathbf{H}, \underline{H}, L)$

For instance, $R_4$ states that in clock cycle 4, the output of scan chain 1 is 'U', the output of scan chains 2 and 3 is 'H', and the output of scan chain 4 is 'L'.

The underlined responses should be masked. This is the case for the unknown responses 'U' in $R_1$, $R_3$, and $R_4$, and for the unreliable response 'H' at scan chain 3 in $R_6$.

The bold responses indicate 'essential responses' on which a fault effect can be observed. The responses $R_1$ to $R_6$ are responses of the fault-free CUT, for a certain test pattern that can detect certain faults. When such a fault is present in the CUT, then one or more of the essential responses will show a different value, and hence the fault is detected. The essential responses should not be masked by the XML.

The remaining responses, which are neither underlined nor bold, are responses that are 'L' or 'H'. These are non-essential responses, and no fault effect is observed on these responses for those faults that are targeted by the test pattern. It may however be the case that other faults, which are not targeted by the current test pattern, do result in fault effects that can be observed on these responses. These faults are however detected by other patterns. Hence, the non-essential responses may be masked without affecting the fault coverage. It nevertheless is preferred not to mask these responses, since multiple detections of faults will lead to better defect coverage.

All responses in $R_1$ should be masked. This is achieved by setting control signal *mask_all* to 1 in clock cycle 1. In clock cycles 2 to 6, *mask_all* is set to 0.

No responses have to be masked in $R_2$ and $R_5$, and hence control signal *mask_enable* is set to 0 in clock cycle 2 and 5. In the remaining clock cycles, masking is required and *mask_enable* is set to 1.

In the other test responses ($R_3$, $R_4$, and $R_6$), the following masking is required:

- $R_{3,2}$ (i.e. the output of scan chain 2 in vector $R_3$) is 'U' and should be masked.
- Fault effects can be observed on $R_{3,3}$, $R_{4,3}$, and $R_{6,2}$, and hence these essential test responses should not be masked.
- $R_{4,1}$ is 'U' and should be masked.
- $R_{6,3}$ is an unreliable response and should be masked.

These conditions can be used in combination with the symbolic expressions for the mask, as given by the output of the phase shifter. The result is the following system of linear equations:

$R_{3,2}$: $x_2 \oplus x_4 \oplus m_1 = 1$

$R_{3,3}$: $x_1 \oplus m_2 = 0$

$R_{4,1}$: $x_1 \oplus x_2 \oplus m_1 \oplus m_2 = 1$

$R_{4,3}$: $x_3 \oplus x_4 \oplus m_3 = 0$

$R_{6,2}$: $x_1 \oplus x_2 \oplus x_3 \oplus m_1 \oplus m_2 \oplus m_4 = 0$

$R_{6,3}$: $x_1 \oplus x_2 \oplus m_1 \oplus m_2 \oplus m_5 = 1$

Such a system of linear equations can easily be solved, e.g. using Gauss elimination. A possible solution is: $x_1 = 1$, $x_2 = 0$, $x_3 = 1$, $x_4 = 0$, $m_1 = 1$, $m_2 = 1$, $m_3 = 1$, $m_4 = 0$, and $m_5 = 0$.

Substituting this solution into the symbolic equations at the output of the phase shifter, and considering the values of the control signals *mask_enable* and *mask_all*, gives the following masks:

$\text{mask}_1 = (1, 1, 1, 1)$

$\text{mask}_2 = (0, 0, 0, 0)$

$\text{mask}_3 = (0, 1, 0, 1)$

$\text{mask}_4 = (1, 1, 0, 0)$

$\text{mask}_5 = (0, 0, 0, 0)$

$\text{mask}_6 = (0, 0, 1, 0)$

The test responses after masking, at the input of the compactor, are now as follows (the symbol ‘m’ denotes a test response that is masked):

$R_1' = (\underline{m}, \underline{m}, \underline{m}, \underline{m})$ ⇒ (1, 1, 1, 1) is the input at the compactor

$R_2' = (L, L, H, L)$ ⇒ (0, 0, 1, 0)

$R_3' = (H, \underline{m}, L, m)$ ⇒ (1, 1, 0, 1)

$R_4' = (\underline{m}, m, H, L)$ ⇒ (1, 1, 1, 0)

$R_5' = (H, L, L, H)$ ⇒ (1, 0, 0, 1)

$R_6' = (L, H, \underline{m}, L)$ ⇒ (0, 1, 1, 0)

The following can be seen:

- All responses that should be masked (i.e. the underlined responses), are indeed masked.

- All essential responses that should not be masked (i.e. the bold responses), are indeed not masked.
- Some non-essential responses that might be masked, are masked. These are indicated by the symbol 'm' that is not underlined.

**End of example**

The example illustrated how to compute the values for the XML control signals. This process can easily be included into an ATPG (automated test pattern generation) tool. Fig. 3 shows an example implementation.

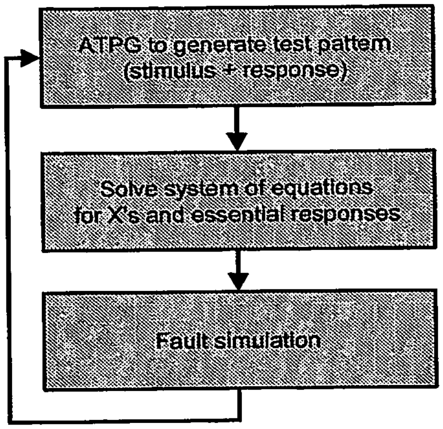
ATPG to generate test pattern
(stimulus + response)
Solve system of equations
for X's and essential responses
Fault simulation

Figure 3: **ATPG flow for computing XML control signals**

The ATPG tool generates a test pattern that detects certain faults ('target faults'). The test pattern consists of the test stimuli and the corresponding test responses. The ATPG tool next analyses the test responses, to identify all tri-state, unknown, and unreliable responses that should be masked, as well as to identify the essential responses that should not be masked. Essential responses are those responses that should be observed in order to detect the target faults.

The values for the *mask_all* and *mask_enable* signals are determined by identifying response vectors in which either all or none of the responses should be masked.

The system of linear equations, derived from symbolic simulation of the XML operation, can next be solved to determine the control inputs for the LFSR in the XML. The system contains equations for all 'X' responses that should be masked, and at least one essential response that should not be masked. If the equation system is solvable, then additional essential responses, which show the fault effects of additional target faults, may be added to the equation system, and the equation system is solved again. This is repeated until all essential responses are added to the equation system or until the equation system is not solvable anymore. The order of adding the essential responses to the equation system may be determined by the number of faults that can be detected by each essential response. First, the essential responses that detect most faults are added to the equation system.

The final step is to do fault simulation, using the test responses after masking, to determine which faults are actually detected. This step is required, since some essential responses may have been masked in case the equation system could not be solved. The target faults that are not detected are targeted by the ATPG again when generating subsequent patterns.

Fig. 3 shows how the XML control signals can be computed per test pattern. An alternative implementation is to compute the control signals by solving the system of linear equations for only part of a test pattern, or for multiple test patterns.

The chance that the system of equations can be solved, is larger if the number of equations is small. Hence, the number of test responses that should be masked and that should not be masked, should preferably be kept as low as possible. For this reason, it is beneficial to have *mask_all* and *mask_enable* control signals, since these allow to mask or not mask complete response vectors, and the corresponding equations do not have to be solved. Similarly, it is also beneficial to not consider the non-essential responses on which no target fault effect can be observed. These non-essential responses may be added to the equation system later, after all essential responses have been added, as long as the system of equations is still solvable.

The proposed XML as described in this invention, provides an effective, flexible, and cheap solution for masking of 'X' responses. The XML allows for reducing the amount of masking data that has to be stored on the ATE, and reduces the bandwidth requirements for transporting the mask data from the ATE to the IC.

The XML can generate a large number of different masks, controlled from the ATE. The masks, and the corresponding control signals to control the XML, can be computed during ATPG. The XML is based on dynamic LFSR reseeding, in which the LFSR state is partially updated from the ATE in each clock cycle. This provides a way to compress the XML control data. For a CUT with $n$ scan chains, only $q$ control bits are needed per clock cycle for reseeding the LFSR, where $q < n$. The LFSR, plus the optional phase shifter with weighting logic, expand the $q$ inputs to an $n$-bit mask. The additional control logic allows to easily mask all scan chains, or to prevent masking per clock cycle.

The XML hardware is simple, and requires very little silicon area when implemented on-chip. The XML may also be implemented partly on-chip and partly off-chip. The number of control signals to control the LFSR state, the size of the LFSR, the size of the phase shifter, the size of the weighting logic, as well as the number of control signals related to *mask_all* and *mask_enable*, can be configured differently per design, depending on the profile of the expected 'X' responses.

CLAIMS:

1. An apparatus used in the testing of an integrated circuit, comprising:

a circuit under test that is part of an integrated circuit;

a compactor to compress test responses from the circuit under test; and

masking circuitry coupled between the circuit under test and the compactor for masking one or more of the test responses from the circuit under test,

characterized in that

the masking circuitry further comprises decompression circuitry for receiving compressed mask data and providing decompressed mask data.

2. An apparatus as claimed in claim 1 wherein, the decompression circuitry further comprises a linear-feedback shift register.

3. An apparatus as claimed in any of the preceding claims wherein, the decompression circuitry further comprises a phase shifter.

4. An apparatus as claimed in any of the preceding claims wherein, the decompression circuitry further comprises weighting logic.

5. An apparatus as claimed in any of the preceding claims wherein, the compressed mask data comprises at least one control signal.

6. An apparatus as claimed in claim 5 wherein, the at least one control signal is a mask all control signal.

7. An apparatus as claimed in claim 5 wherein, the at least one control signal is a mask enable control signal.

8. A method used in the testing of an integrated circuit, characterized in that it comprises the steps of:

providing compressed mask data to decompression circuitry;

decompressing the compressed mask data to produce decompressed mask data; and

masking test responses from the integrated circuit in response to the decompressed mask data.

9. A method for computing compressed mask data for use in masking test data from an integrated circuit, characterised in that it comprises the steps of:

generating a set of equations associated with the mask data; and

solving the equations to obtain compressed mask data.